

## II. CLAIM AMENDMENTS

1. (Currently Amended) A method for dynamically binding a program module in a terminal, in which one or several programs are running, ~~said and in which method comprising:~~

~~storing~~ subroutines ~~are stored~~ in said program module,

~~providing the~~ program module ~~is provided with a~~ subroutine name and parameters ~~first tag,~~

~~wherein to starting the~~ binding, ~~by the~~ program making ~~makes a~~ call to a subroutine,

~~and supplementing the call is supplemented with~~ subroutine name and parameters ~~first call data~~ to select the program module for binding, in which the called subroutine is stored,

~~wherein supplementing the~~ subroutine name and parameters ~~first tag is supplemented with an~~ auxiliary second tag,

~~supplementing the call is also supplemented with a~~ public key second call data, ~~in connection with the binding,~~

comparing said subroutine name and parameters ~~first tag~~ stored in the program module ~~is compared with the~~ subroutine name and parameters ~~first call data~~ transmitted in the call,

~~and comparing the~~ auxiliary second tag ~~is compared with the~~ public key second call data transmitted in the call, and

~~selecting the program module to be bound is selected to be the program module which matches with the first-call subroutine name and parameters data and the public key second-call data transmitted in the call.~~

2. (Currently Amended) The method according to claim 1, wherein the auxiliary second tag to be formed in the program module contains a digital signature.

3. (Currently Amended) The method according to claim 2, further comprising verifying wherein the second-call data are supplemented with a public key, on the basis of which the digital signature of the public key second-call data formed in the program module is verified.

4. (Currently Amended) The method according to claim 1, further comprising storing wherein the auxiliary second-tag to be formed in the program module is stored in an encrypted form.

5. (Cancelled)

6. (Currently Amended) The method according to claim 1, further comprising storing in which the program module is stored in a server communicating with a digital network, wherein the terminal used is a mobile terminal, and performing the binding of the program module is performed at least partly by messages complying with the WAP protocol.

7. (Currently Amended) A terminal comprising:

means for dynamically binding a program module, in which program module contains at least one stored subroutine and a subroutine name and parameters first tag, and which terminal also comprises

means for running programs,

means for starting binding by performing in the program a call to a subroutine, the call being supplemented with subroutine name and parameters a first call data to select that program module for binding in which the called subroutine is stored, wherein the program module contains a stored auxiliary second tag; ~~the terminal also comprises~~

means for adding a public key second call data to the call,

means for comparing said subroutine name and parameters first tag stored in the program module with the subroutine name and parameters first call data transmitted in the call,

means for comparing the auxiliary second tag with the public key second call data transmitted in the call, and

means for selecting a program module to be bound on the basis of said comparison,

wherein the program module to be bound to be the program module which matches with the subroutine name and parameters data and the public key transmitted in the call.

8. (Currently Amended) The terminal according to claim 7, wherein the auxiliary second tag formed in the program module contains a digital signature.

9. (Currently Amended) The terminal according to claim 8, wherein ~~the second call data are supplemented with a public key,~~ on the basis of said public key which the

digital signature of the ~~second~~auxiliary tag formed in the program module is arranged to be verified.

10. (Currently Amended) The terminal according to claim 7, further comprising means for binding a program module stored in a server communicating with the Internet network, wherein the terminal is a mobile terminal, and said mobile terminal comprises means for performing binding of the program module at least partly by messages complying with the WAP protocol.

11. (Currently Amended) A method in a terminal, in which one or several programs are running, said terminal comprising at least one program module, which is provided with a subroutine name and parameters ~~first~~-tag and an auxiliary ~~second~~ tag, and said at least one program module comprises at least one subroutine, wherein the method comprises:

making a call by the program to a subroutine;

supplementing the call with first call data to select a program module for dynamically binding among said at least one program module, in which the called subroutine is stored;

supplementing the call ~~is also supplemented with a public key~~ second call data, in connection with the binding;

comparing said subroutine name and parameters ~~first~~-tag stored in said at least one program module with the subroutine name and parameters ~~first call data~~ transmitted in the call;

comparing the auxiliary ~~second~~-tag with the public key ~~second call data~~ transmitted in the call; and

selecting the program module for binding to be the program module in which the subroutine name and parameters ~~first~~-tag matches with the subroutine name and

parameters first-call data and the auxiliary second-tag matches with the public key second-call data.

12. (Currently Amended) A terminal comprising:

a binding server for dynamically binding a program module, which program module contains at least one subroutine, a subroutine name and parameters first-tag and an auxiliary second-tag;

an element for running programs and configured to:

perform in the program a call to a subroutine,

supplementing the call with subroutine name and parameters first-call data to select that program module for binding in which the called subroutine is stored;

supplementing the call with a public key second-call data;

comparing said subroutine name and parameters first-tag stored in the program module with the subroutine name and parameters first-call data transmitted in the call;

comparing the auxiliary second-tag with the public key second-call data transmitted in the call; and

selecting a program module to be bound on the basis of said comparison,

wherein the program module to be bound to be the program module which matches with the subroutine name and parameters data and the public key transmitted in the call.